AUSTRALIAN MICRO NEWSLETTER

KAOS

FOR PEOPLE WHO GOT SMART

SECRETARY.....JOHN WHITEHEAD
MEMBERSHIP....RALPH HESS.....
RECORDS.....RON CORK.....
EDITOR.....BRIAN BUSBY....
EDUCATION.....JEFF KERRY.....
AMATEUR RADIO.....VK3BUS...
CLIVE HARMAN....
HARDWARE.....DAVID ANEAR....
SOFTWARE.....JEFF RAE.....

ADDRESS ALL CORRESPONDANCE TO :-
SECRETARY,

AIM-OSI-APPLE-MAC-IBM-RABBLE 65-C64-VIC20-PEACH-UK101-BBC

REGISTERED BY AUSTRALIA POST
PUBLICATION No. VBG4212

Vol.5 No.11 August 1985.

Kaos is branching out into new territories, no longer are we tied to OSI, or even 65Ø2. There are now several different computer groups operating within the club and interesting information is being printed on a range of computers. We regard ourselves primarily as a hardware and software hackers group. If you are a former member and are still interested in computers, you may find it worthwhile to rejoin. We still meet at the Essendon school - you are welcome!

Notice header no.2! We will print a new one each month for a short while and then choose the best by popular vote. So please continue to give your opinions and ideas.

At the back of this issue, there is an enrolment form. No, it's not yet time to pay your sub; this form is for new members.

The club superstructure will, hopefully, be sorted out at the August meeting. So please come and help settle things. I hear that there is to be a Junk Sale at the September meeting, so start dusting it off. (I'm looking for old data books and a UV lamp - what do you want?)

We want to publish some photos shortly. If you have any interesting ones, please supply as black and white prints, preferably around 5"x4".

Back copies of Kaos Newsletter are available from Ron Cork at $1 a copy.

The next club meeting will be at 2pm on Sunday 25th August at Essendon Primary School , Raleigh St. All welcome. Deadline for newsletter articles etc. is Friday 13th September, but note that we seldom get a postal delivery on Fridays!

INDEX

SIMPLE APPLE I/O PORT - BY R.WOODHOUSE

The accompanying circuit diagram (Fig 1) is a simple, easy and cheap way of gaining several I/O lines for use with your APPLE][computer.

As the computer is already equipped to handle I/O signals through its GAMES connector, it seems wasteful not to use these.

All lines, both in and out, are fully buffered so as to provide protection to the computer, a very necessary thing.

The I/O port was built on VERO-BOARD using sockets, with a trailing lead and 16 pin D.I.L. plug to connect with the computer games socket and two 16 pin D.I.L. sockets for both IN and OUT.

Power for this project (+5V) is available from the computer (pin 1) as is GROUND (pin 8).

Once the board is complete and checked for possible shorts, it may be plugged in, in lieu of the games controller and checked, using a simple BASIC program and test circuit.

L.E.D.'s and 5V reed relays may be used direct from the O/P ports and if a decoder such as a 74LS47 is used a 7 segment L.E.D. display may be used.

A simple audio generator using a 555 timer chip (to be described later) , may be used and operated from the four O/P lines to produce sounds and even music that is very difficult with the APPLE's sound system.

Inputs may be used from almost any source as long as they are at TTL levels (0 to +5V d.c.) and this could concievably be used for burglar alarms, model train controllers, temperature measurement etc.

In fact the possibilities are very nearly endless once your I/O port is operating. It only requires some patience to write the appropriate software and a minimal amount of hardware.

PARTS LIST

2 X 74LS08 I/C's
3 X 47Kohm .5W resistors
2 X 14 pin D.I.L. sockets
2 X 16 pin D.I.L. sockets
1 X 16 pin D.I.L. plug
1 metre of 10 conductor flat cable

The accompanying circuits (Fig 2), and the test programs that go with them are for testing the I/O port when completed.

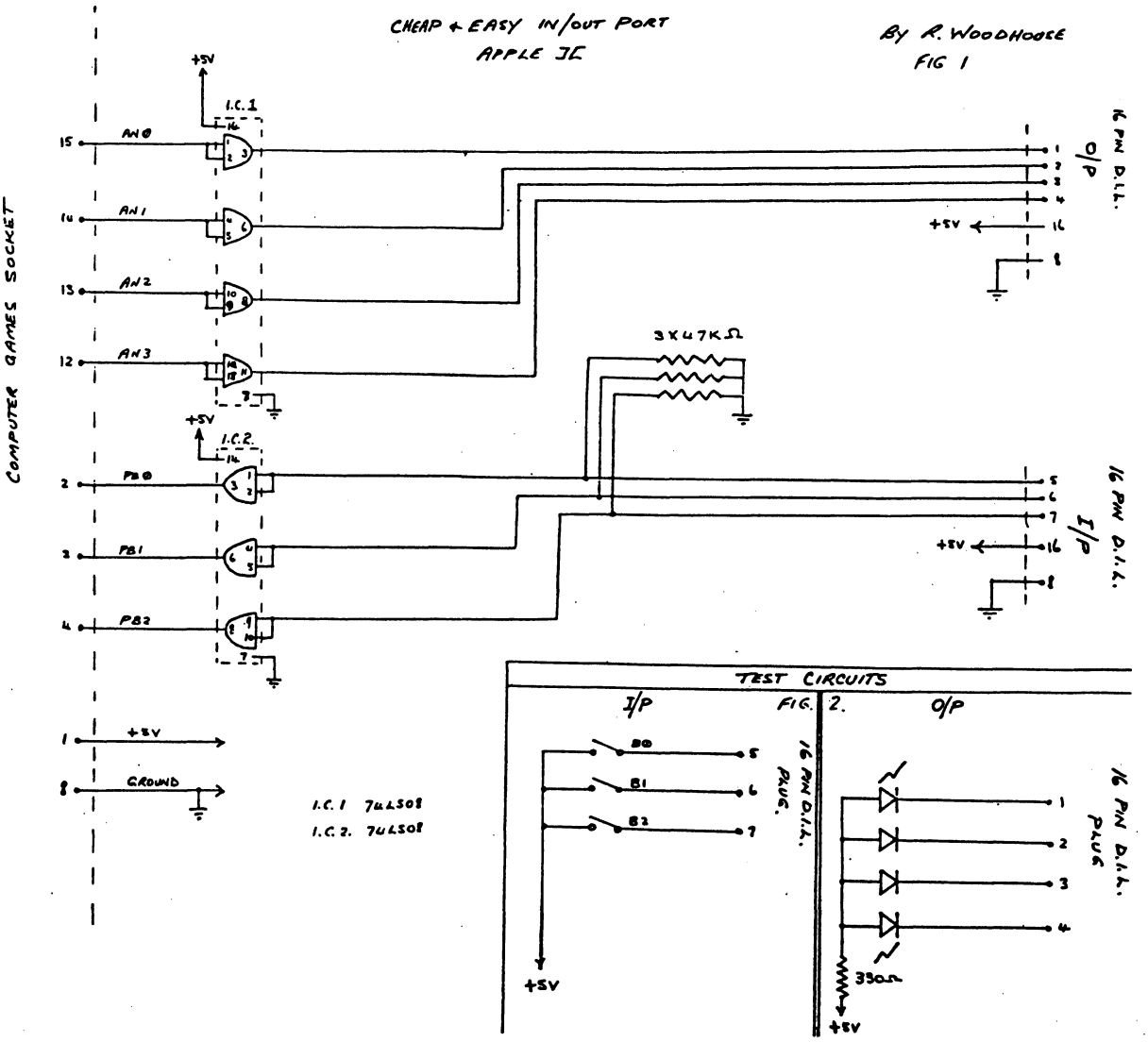
The O/P test turns all L.E.D.s off and then turns them on in sequence at an approximate 1 sec. rate. When all are lit it waits 1 sec. and then starts over again.

The I/P test displays "All switches open" until at least one switch closes. It then indicates which switch or switches are closed.

TEST CIRCUIT PARTS LIST

2 X 16 pin D.I.L. plugs
4 X L.E.D.s
1 X 330 ohm 1W resistor
3 X S.P.S.T. push button S/W (momentary contact)

This I/O port is utilised for the Sound Generator published last month. Connector numbers correspond in the circuit diagrams...Ed.



THE MEETING WAS KAOS

Arriving late, it was gratifying to see lots of cars parked in the grounds, and lots of people, with a sprinkling of new faces, around the computers and at the meeting.

Our usual M.C., Ron Cork, was not present, as he had just come out of hospital after a minor operation (rumour hath it that there was some sort of input port problem!). So David Anear had to come to the rescue and conduct.

KAOS is in process of rejuvenating and changing its image from a 6502 based club to one where the number on the CPU does not matter as long as your interest is new frontiers in hardware and/or software.

As part of this, Brian Busby, the Editor of KAOS, is organising a competition for a new image cover for the magazine. For further details see this issue, p.1 .

Ray Gardner assures us that his Universal Intelligent Industrial Controller board has arrived and will be available at the next meeting. Also the Forth course, based on the use of this board, is definitely on. Again, details at the meeting, as they are not yet finalised.

THE CPM FILE

The perpetrator of this article regrets that due to an unfortunate combination of circumstances, to wit

(a) The untimely death of one of his 8" drives,
(b) The imminent marriage of his elder daughter, and
(c) The urgent need to finalise the 1985 Tax Returns

has necessitated the postponement of this month's article.

Sorry, but that's death and taxes (and marriage too) for you. Next month for sure. Bye. rmh

COMPUTERS IN EDUCATION - WHAT'S GOING ON ?

by Jeff Kerry

Things have come a long way in the past four or five years. With elaborate microcomputers now commonplace, the widespread changes everyone was expecting to happen to education, are actually happening - you can tell, because they're now regarded as almost routine ...

No longer just for hackers and enthusiasts, personal computers have become almost as common a sight around schools as chalkboards, videos and other equally old-fashioned technology. The situation at the high school where I teach seems typical.

Only a few years ago, the first of us enthusiasts were patching together our own, cassette-based, wonder machines. Nowadays, many of the teaching staff are "computer literate", there are networked classroom systems, and courses running at all grade levels in the school. What's more, the students' Computer Club is the second most popular student activity in the school (guess what the first is!) and some kids have more disks than books in their lockers.

You hear a lot about the so-called "technology lag", that social changes take years to filter through to the school system.

Not so. Even the Education Department here in Victoria now recommends that ALL students, from preps through Year 12, have continuous access to word processing equipment. Teachers of English haven't been all that slow to respond. Kids will edit their own work much more happily on a WP than if written out laboriously by hand. Databases are getting used in History classes, teachers of Art and Graphics can be heard muttering about Mice and Plotters, Science experiments can feature computer interfaces, and so on.

The biggest change going on at present is the amazing sophistication of the new software. Programs that you wouldn't have believed would run on an ordinary old Apple II. Typically, they make terrific use of fast color graphics, sound and peripherals, and teach concepts in ways you just wouldn't have imagined until you'd seen them. If you've seen programs like "The Factory", you'll know what I mean.

If you want to make money writing software to sell to schools these days, you'll need to be thoroughly familiar with the appropriate machines (most schools have Apples), and you'll probably need to be a graphics whiz, and with a fair idea of schools' curriculum requirements. (Nothing worse than a Super New Educational Program that turns out to be either too simple or just irrelevant.)

It's hard to realize that these versatile machines, which we now take so much for granted, just weren't on the scene only a few years ago. The next few years should be most interesting ...

Accessing Viatel by Rodney Eisfelder

Viatel is Telecom's national Videotex system. It can be used to access information on a wide variety of topics from your own bank balance to countries where Yellow Fever is endemic. It can also be used to order goods and services and down load software.

Unlike Bulletin Boards and services such as the Australian Beginning, Viatel requires more than a "dumb teletype" to be usefully used.

To use Viatel you need a 1200/75 baud modem (you send at 75, Viatel sends at 1200). Your micro-computer must attempt to emulate the Prestel Terminal specification which has a screen of 40 columns by 24 lines. The cursor wraps from the end of one line to the start of the next and from the bottom of the screen to the top. There is no scrolling. Eight colors are defined (although many implementations ignore the problem of color). Mosaic graphics (in which each character cell is split into a 2*3 grid) are an essential part of any emulator. The full specification defines two types of mosaic graphics, one in which the graphics blocks are contiguous and another in which they are separated by the background color.

Flashing characters are also part of the specification. Attributes (such as color, flashing, graphics mode) are reset to the default values (white on a black background with no flashing or graphics) at the start of each line.

(continued page 18

S U P E R B O A R D ©August 1985

Newsletter of the Ohio Superboard User Group, 146 York Street, Nundah 4012.

ADVENTURES AND THE OSI

An introduction to Adventure:

Adventure games have been played on computers of all types for many years and are one of the most difficult games to play, and certainly the hardest to create. Essentially the player is in a universe of the writer's imagination, questing for a goal which is often obscure, and having to solve problems which should have logical solutions, but sometimes don't. Usually the objectives are to survive, and find some sort of treasure. The location can be caves, castles, outer space, or even in open surroundings.

The first adventure was simply titled "Adventure" and was written in Fortran, to run on a DEC PDP-10 computer with 300k of memory. Of course the introduction of the microprocessor meant that adventures had to be crammed into much smaller memory, usually 16k. Much of the magnificent wording which described rooms in the original Adventure had to be left out. An example of such wording follows:-

You're at a low window overlooking a huge pit, which extends up out of sight. A floor is indistinctly visible over 50 feet below. Traces of white mist cover the floor of the pit, becoming thicker to the left. Marks in the dust around the window would seem to indicate that someone has been here recently. Directly across the pit from you and 25 feet away, there is a similar window looking into a lighted room. A shadowy figure can be seen there peering back at you. What now?

This is nowhere near the longest room description in Adventure, but such descriptions could not possibly be used in even a 64k machine. The IBM, of course, offers such possibilities. Other machines could possibly call in the description of the rooms from disk, however most adventures for home computers merely truncate the description drastically to only the most essential details.

Probably the most advanced and complex adventure game is ZORK, written entirely in compiled code. While ZORK doesn't have enormous room descriptions, it does accept almost any answer. ZORK was also written on a PDP-10, and is usually supplied on 2 to 3 disks, which says something of it's size. ZORK has it's own interpreter, just like a BASIC interpreter, which makes it easier to adapt to different processors. With ZORK, you can say "Take the bomb and put it at the foot of the door". Almost all other adventures would require "Take bomb","Put bomb", *WHERE?*,"Door".

Of course ZORK has already been eclipsed by graphical adventures and also role playing games typified by Dungeons and Dragons. The ultimate adventures will come when the Laser video disk is coupled to home computers. You will then see the rooms through your character's eyes. You will also be able to select your character's traits and so the adventure can be different every time you play it, the final outcome depending on the role you have adopted. With varying strengths of physical and intellectual capacity, several million different characters would be possible. A strong heart would also be recommended for the player. To see yourself about to be destroyed would provide quite a shock. The psychiatrists might do well out of it!

However we will have to wait for this. For the moment we will be limited to simple 8 or 16k adventures for the OSI. Although several quite good 8k adventures have been written,(even 5k ones!) I really think 16k is more appropriate. A really good adventure should have perhaps 40 or more rooms, and this is simply not possible with 8k.

Solving Adventures:

There are two cardinal rules to observe when setting out on a new adventure, The first one is to look at everything, and the second is to draw a map as you travel. Most objects you come across will have some role to play, and most will have only one role, though this is never certain. With the OSI adventures, you won't find many red herrings or dead ends, simply because the 8K memory doesn't allow any space for it. However in 16K games, you will find routes which lead absolutely nowhere, and objects which have not the slightest use except to annoy you and delay the solving of the puzzle.

Drawing up a map will always enable the adventure to be solved much faster, as it prevents random wanderings over the same ground. On your map, you should name each room and mark the contents as you first find them, and also note the exits. Wherever you start drawing your map on the paper will almost certainly be the wrong place, so to avoid crunching up the last part into some obscure corner, have a second sheet ready to stick on. Some adventures have one-way movement which is rather hard to represent on a map. Perhaps a different colour pen might help there. If your adventure contains anything which suggests a maze, you should most carefully document your journey. This will save much wandering in a later game when you meet with that inevitable nasty fate in early games.

Some games have ramdom distribution of objects as in our Treasure Quest game in KAOS, however most real adventures have a fixed and logical method for solving the puzzle. If you encounter a problem, you will not be able to solve it without the correct object. Sometimes you will not be able to return to get it, and have to replay the game over. Some adventures have a "save the game" feature, though I haven't seen one for OSI. This enables you to recall a partly completed game, and is a very useful thing to do before some heroic but risky venture, such as attacking a dragon!

Our Adventure:

The adventure I will begin next month will run on almost any computer with 8K and Microsoft Basic. It is based on an adventure written by Sean Davidson for our last programming competition. Sean's adventure, titled "Vampire's Treasure" used peeks and pokes to page 2 which is fine for OSI, but wouldn't suit all the computer types now in KAOS. For this reason it has been re-written to use variables instead. In doing so, I changed objects, rooms and altered the scenario to suit, though it remains very much Sean's Adventure. Following this adventure I may consider doing one in 16K which will have all the bells and whistles. After Vampire's Treasure, a map will be printed in a following KAOS, and I will explain just how it all works. Much of this has already been covered in Treasure Quest. Like any other adventure, the re-worked Vampire's Treasure may contain a bug or two, though I am very careful. If so, I'd like to hear about it.

ABS FUNCTION

B=ABS(A) is a mathematical function that isn't used often. It simply makes B's sign positive. At a glance you wouldn't think this would be so handy, but it can be most useful.
This line will detect a positive integer between 20 and 80 inclusive:-

```
20 IF INT(N)<>N OR ABS(50-N)>30 THEN PRINT"Wrong!"
```

The general formula is ABS((A+B)/2-N)>(B-A)/2 where A is the smaller number. This line will check an approximately correct answer. A is the input answer while R is the right answer. E is the absolute error, and P is the more important Percentage or relative error.

```
40 E=ABS(A-R):P=100*E/R:IF P<5 THEN PRINT"Within 5%"
```

Ed Richardson.

RABBLE OZI COMPUTERS FORTH BOARD

104 MACINTOSH STREET, SHEPPARTON, VICTORIA 3630, AUSTRALIA

The FORTH BOARD has been designed for those one-off dedicated applications, either in industrial or hobby applications. The board uses the ROCKWELL R6511AQ microprocessor. So lets start with a brief description of the R6511AQ.

The 6511 series of microprocessors are packaged in a 64 pin QUIP package, that is staggered 0.05 inch pin spacing. An on chip block of 192 bytes of ram is used for page zero and the stack. Also on chip is a flexible I/O structure that allows for 32 I/O lines which may be programmed as inputs or outputs. A software controlled serial port can run up to 68Kbits/sec, the serial port can easily be arranged to cater for networking using the wake-up bit in the SCCR For more information you should obtain the ROCKWELL publication "Document No. 29651N36" Product description order No. 2133 rev 3. this document covers all the variuous features and operating modes of the R6511Q, and R6500/13 microprocessors.

One of the features worthy of mention here is the new instructions that have been added, these are the bit manipulation instructions that allow you to set, reset and test single bits on any port or page zero location.

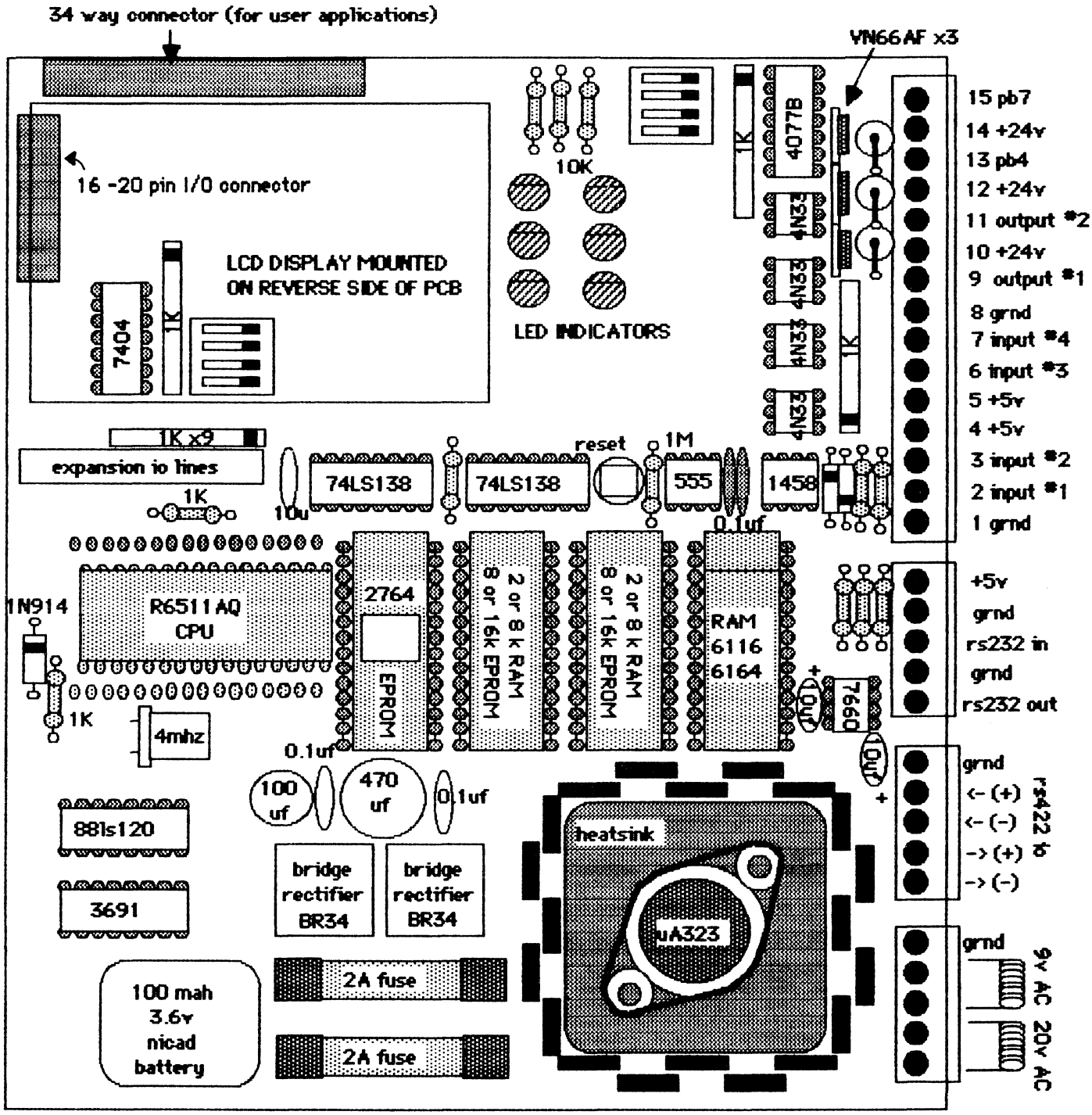
The rockwell forth kernel has been extensively reworked and cleaned up, the kernel rom now includes a stand alone monitor which includes memory examine/alter and dissassembler.

In addition to the 16/20 pin OHIO standard I/O bus the board provides for 4 opto-isolated inputs and 4 buffered 12/24 volt outputs. Oh, and yes a 16 character by 2 line LCD display can be mounted on the rear of board. Nicad battery backup is provided for the main ram and the CPU ram. If you refer to the component overlay on the next page you can probably fathom out the rest yourself. Bare boards with minimal documentation will be ready in time for the AUGUST K.A.O.S meeting. Cost $25-00 including the dreaded sales tax. Get in early !!

Ray. Gardiner july 1985

COMPONENT OVERLAY

Rabbit Ozi Computers Forth dedicated controller



AUTHORS! Please try to form your articles and diagrams to fit the Kaos page (170x260mms. or 6.5"x10"). Narrow text wastes space. Make the pages full length to avoid "For Sale" and other notices in the middle of your composition...Ed.

My Superboard Part 11
by John Whitehead

Some months ago I bought a second hand Shugart SA400 disc drive from George and came by an unpopulated Tasker disc driver board. I had to go to six shops around Melbourne before I had a full set of ICs for the board.

I have had it running for a week using HEXDOS and am now learning about the software. Sofar I have a program that allows me to load and save a source listing of a 6k m/code program into D Dodds Assembler that I have in EPROM. But this article is about what I have learned sofar on hardware.

I was informed I needed a data separator. I obtained one that used the MPI B51 circuit (fig. 1a). This I fitted to the Tasker board. To enable me to work out what a data separator does, I worked out the waveforms for the data combiner circuit that puts the data and clock on the disc as fig. 1b and 2.

For the cable between the drive and the Tasker board I used 34 way flat ribbon with the board end separated and connected as fig 3. Connection 27 on the board goes to the line load resistors then to the data separator input.

The Tasker board monostable settings are explained in KAOS Oct 81.

When I powered up the system, I found I had to ground the motor control signal (pin 16 at drive) to energise the motor and that the head load solenoid was always on. On the drive is a 14 pin socket for selecting options. I found by experiment that :-

MH = head load with motor on
. = not marked
MX = dont know
DS3 = select drive as # 3
DS2 = select drive as # 2
DS1 = select drive as # 1
HL = head load on power up.

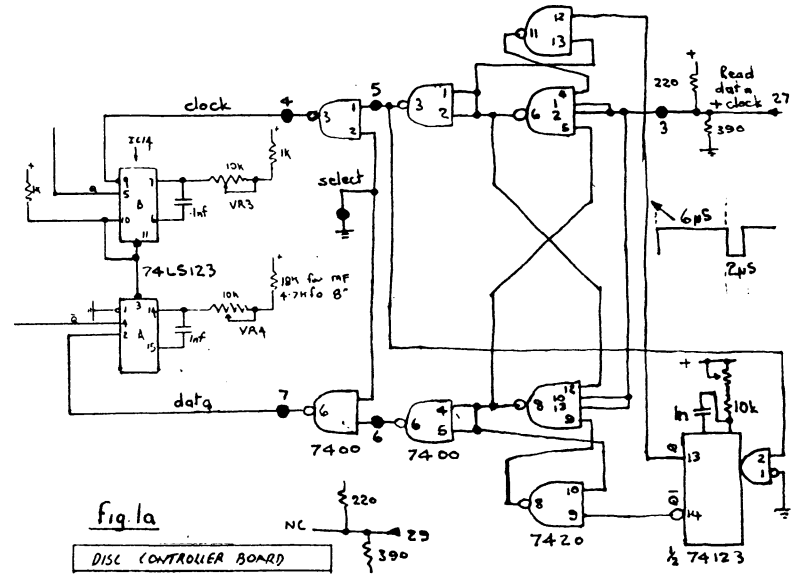


Fig 1a

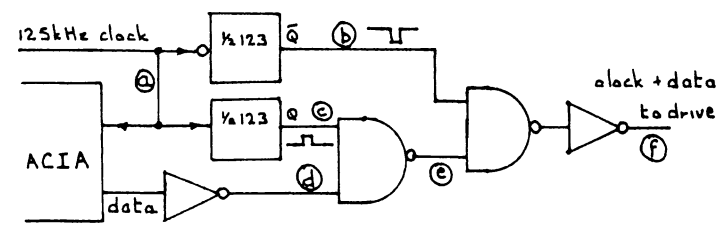


Fig 1b

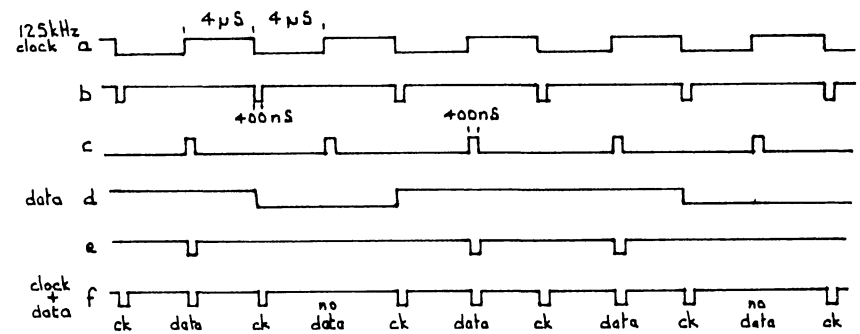


Fig. 2

By removing the HL strap and putting it in MH, I found the head loads when the motor runs.

Now, how to control the motor ?. There were some ideas in the March and April 1985 KAOS. However, with the Tasker board, it is unnecessary to cut tracks since mods are easy on the board. I have used a cicuit by Paul Dodd in KAOS May 82 and fed its output to the input of the 'fault reset' line driver. This gets turned on whenever the PIA at $CØØØ to Ø3 or the ACIA at $CØ1Ø to 11 is accessed and keeps the motor going for about 2 sec. after access ends. I used the spare half of the 74123 on the data separator for this as fig 4. I have not used any circuit to block off the index hole while the motor gets up to speed as mentioned on page 6 of KAOS April 85 and have had no problems. (Does the drive have this circuit built in ?)

I use my superboard at 2MHz for most programs except games so have connected my Tasker board to give a 125kHz disc clock with a 2MHz input (see fig. 5).

The Shugart drive track stepper motor is slower than an MPI drive, so one byte in software needs changing to increase the delay time between step pulses. The same byte is also altered to enable me to run at 2MHz but does not have to be altered back to run at 1MHz. The drive also has only 35 tracks.

Sofar I have only run Hexdos which does not divide tracks up into sectors. With other disk systems more timing loops are used due to sectors. Do any of you veteran disk users have a solution that will enable me to manually switch my superboard from 2 to 1MHz. so that all timing and clock pulses are automatically altered.

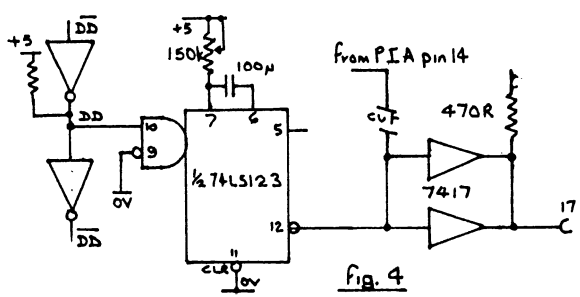


Fig. 4

	+5	36	
	+5	35	
	+5	34	
		33	33 ØV
	ØV	32	32
SELECT DRIVE 2		31	31 ØV
	ØV	30	30 READ DATA
SEP. DATA		29	29 ØV
	ØV	28	28 WRITE PROTECT
SEP. CLOCK		27	27 ØV
	ØV	26	26 TRACK Ø
SELECT DRIVE 1		25	25 ØV
	ØV	24	24 WRITE ENABLE
WRITE DATA		23	23 ØV
	ØV	22	22 WRITE DATA
HEAD LOAD		21	21 ØV
	ØV	20	20 STEP
LOW CURRENT		19	19 ØV
	ØV	18	18 STEP DIR.
FAULT RESET		17	17 ØV
	ØV	16	16 MOTOR CONTROL
RESET STEP		15	15 ØV
	ØV	14	14 DRIVE 3 SEL.
STEP IN		13	13 ØV
	ØV	12	12 DRIVE 2 SEL.
ERASE ENABLE		11	11 ØV
	ØV	10	10 DRIVE 1 SEL.
WRITE ENABLE		9	9 ØV
	ØV	8	8 INDEX HOLE
INDEX HOLE		7	7 ØV
WRITE PROTECT		6	6
READY DRIVE 2		5	5 ØV
SECTOR		4	4
FAULT		3	3 ØV
TRACK Ø		2	2
READY DRIVE 1		1	1 ØV

TASKER BOARD FIG. 3 DISC DRIVE

I HAVE NOT USED SEP. DATA, AND SEP. CLOCK IS USED FOR CLOCK + DATA

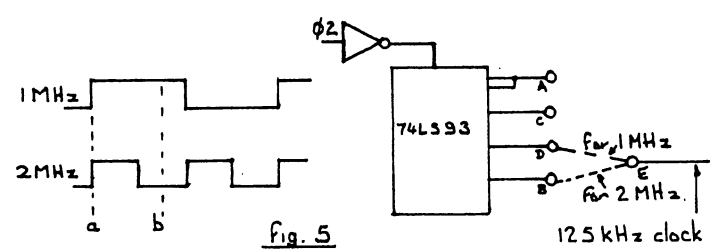
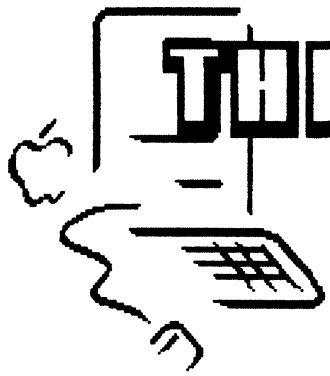


Fig. 5



THE MACINTOSH

OR HOW TO GET HOOKED ON WINDOWS

BY David J. Anear

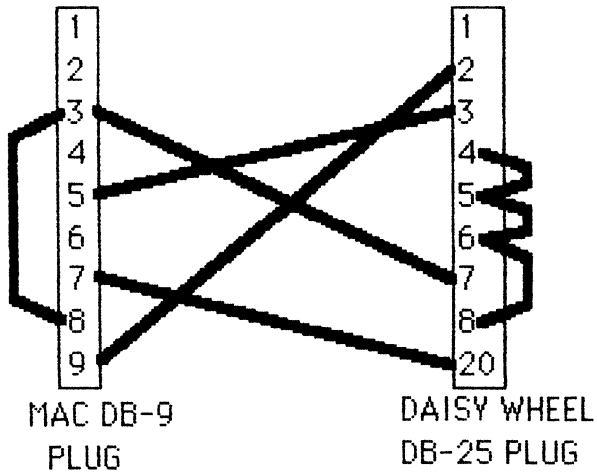
The Apple Macintosh is now 18 months old, and still there is very little written about the hardware and especially the varios ports available at the rear of the machine. It was not untill I purchased a copy of INSIDE MACINTOSH, the software developers guide for the Mac that any information about the pin-outs for the two serial ports was obtained. Both serial ports are identical and come out on a DB-9 socket, as does the Mouse connections.
Here are the pin-outs for the ports :-

Pin No	Designation	Printer Cable	Modem Cable
1	GROUND	NO CONNECTION	NO CONNECTION
2	+5 VOLTS	NO CONNECTION	NO CONNECTION
3	GROUND	PIN 7	PIN 7
4	TXD +	NO CONNECTION	NO CONNECTION
5	TXD -	PIN 3	PIN 2
6	+12 VOLTS	NO CONNECTION	NO CONNECTION
7	HSK	PIN 20	NO CONNECTION
8	RXD +	PIN 7	PIN 1
9	RXD -	PIN 2	PIN 3

NOTES :- 1) For RS-232 use Ground, TXD-, and RXD-. RXD+ is grounded.
2) HSK is the Handshake I/P or the EXT CLOCK I/P depending apon the 8530 serial chip mode selection. For RS-232 it is usually connected to pin 20 on the DB-25 plug/socket.
3) +12 volts is for power on detection only.
4) TXD+ and RXD+ are used for RS-422/423 compatability.

For example if you wish to connect a Daisy Wheel like the Brothers 15 to the Macintosh you would use the following connections:-

MACINTOSH TO DAISYWHEEL CONNECTION



The serial ports use a Zilog 8530 Serial chip and this chip is capable of any baud rate up to 50 kilo baud using the Macs internal clock or up to 1 mega-bit rate using an external clock fed into PIN 7 (HSK).

MACINTOSH MOUSE PORT

This is the other DB-9 socket on the back of the Mac and is used for the connection from the single button mouse. It is really the only input which handles TTL levels that can be seen by the Mac at any time without a lot of software overhead. The Mac expects a mouse at this port and stores any button presses or mouse movements as events which can be replayed later to control any program

Macintosh Mouse Port Pin-outs

PIN No.		DESCRIPTION	CABLE COLOUR	MAC.CHIP.PIN
1	-	GROUND (N.C.)		
2	-	+ 5 VOLTS	RED	
3	-	GROUND	BLACK	
4	-	LEFT (X2)	BROWN	6522/PB4
5	-	RIGHT	ORANGE	8530/DCDA
6	-	NO PIN IN SOCKET		
7	-	SWITCH INPUT	YELLOW	6522/PB3
8	-	DOWN (Y2)	GREEN	6522/PB5
9	-	UP (Y1)	BLUE	8530/DCDB

NOTES:- (1) Switch input is normally high (+5v) and is pulled down to ground by pressing button

(2) MAC.CHIP.PIN shows chip type and its pin designation on logic board which that pin terminates at.

THE POWER OF VISICALC PT2

by Bob Geels.

As promised last month, here is a spreadsheet for calculating your Mortgage interest, principal reduction and total interest. It uses an @IF function to establish final payment and is adjustable for monthly or quarterly repayments. Type:

```
/CY
>A110
/-=
>A10
/-=
>A7
/-=
>A5
/-=
/R.A110
B5.D5
>B1
MORTGAGE
>C1
AMMORTIZA
>D1
TION
>A3
INITIAL
>A4
PRINCIPAL
>B3
INTEREST
>B4
" RATE PA
>C3
PAYMENTS
>C4
PER ANN
>D4
REPAYMENT
>A9
PRINCIPAL
>B9
" INT/PAY
>C9
PRIN RED
>D9
TOT INT
```

Three global commands are needed in this spreadsheet to make it work efficiently. Type:

```
/GOR  (Calculate rows to stop back to front referencing)
/GRM  (Manual calculation, type "!")
/GF$  (Dollars and cents format)
```

These commands are saved when you save a spreadsheet to disk. Note:- The input values of the mortgage in this case are along row 6. @NA should be used in these positions and a calculation performed before saving to disk to save memory. Now for the formulae. Type:

```
>A11
+A6
>B11
((B6/C6)*A11)/100
>C11
@IF((D6-B11)>A11,A11,(D6-B11))
>D11
@IF(A11=0,@NA,(D10+B11))
>A12
+A11-C11
/R
A13.A109
RR
>B11
/R.D11
B12.B109
```

Now slowly type: NNR NRRRNR RRR

You should now have a page full of formulae. Goto row 6 and enter your mortgage values. Type "!" and all your payments will be calculated. At the point where your morgage is payed off, @NA will be printed. The final payment is the sum of the last figure in column A and the last figure in column B.

Try typing /IR half way down your calculations and watch the formulae re-adjust themselves as you insert a blank row. this can be useful to seperate your calculations into years. /DR to delete rows.

The @LOOKUP Function

Apart from being able to initiate a sequence if a required value is calculated, the @LOOKUP function can save a great deal of typing in costing applications by coding the values of items and using the @LOOKUP function to select the applicable values as required. i.e. Select the code number of a certain material description and it's value would be brought into calculation. This spreadsheet will demonstrate its capabilities. Clear the screen with /CY. Type:-

```
>A20                          >A3
/-=                           ITEM 1
>A2                           >A4
/-=                           ITEM 2
/R.A20                        >A5
B2.E2                         ETC..
>A1                           >B3
DESCRIPT                      1+B2
>B1                           >E3
CODE NO                       @L
>C1                           D3,B3.B19)
VALUE                         >B3
>D1                           /R.E3
ENT CODE                      B4.B19
>E1                           RRNN
SEL VALUE
```

Now enter a series of descriptions in column A and their values in column C. The values of each item can now be selected at will in any order and any number of times merely by selecting its relevant code number. The @LOOKUP function will look for the code number in column B and select the appropriate value to the right of the code number. This format can easily be suplimented with item quantities in column F and multiplying E * F for for totals in column G.

Note: Leave one row with a 0 value to be selected when no more entries are required. This prevents a result of @NA in the totals.

LOGIC OPERATORS

Visicalc logic operators include @IF, @AND, @OR, @NOT, @TRUE and @FALSE. Anyone who has dealt with Boolean algebra or basic, will know that arguments, (A=B), are reduced to a logic level of 1 or 0, before being looked at by the logic operator. The same applies to Visicalc eccept that the logic operator preceeds the arguments which must all be enclosed in parenthesis and separated by comma's. Also the @IF function can be used to direct calculation in either of two paths depending on the result of the argument. i.e.

@IF((Argument),(This section for a 1),(This section for a 0))

Logic operators may be nested " @IF(@IF(a,b,c),B,C) " in any section of the formula and any combination of logic operators may be used. This, however, is where the parenthesis nesting of arguments and preceeding logic operators gets tricky. Mistakes usually result in an @ERROR message. For starters, we'll put some variables in column A. Type:-

```
/CY                    6
1                      >A7
>A2                    7
2                      >A8
>A3                    8
3                      >A9
>A4                    @NA
4                      >A10
>A5                    @TRUE
5                      >A11
>A6                    @FALSE
                       >B10
```

At B10 Type:- @IF(@AND(A1=1,A2=2),A10,A11)

Note that the argument has sub arguments enclosed by parenthesis, separated by commas and preceeded by their logic operator @AND. This formula asks that A1=1 and that A2=2 before the value of A10 is shown, otherwise the value of A11 is shown. Try changing the values in A1 and A2 and follow the logical conclusion for the answer. Try changing the logic operator to an @OR function and follow the logic again.

Multiple (Nested) Logic Operators

When mixing logical operators, a sequence of priorities must be established. The highest priority preceeds other arguments. i.e. In the following example, a "1" is needed in A1 before other factors (A2,A3) are considered. With the original numbers in column A. Type:-

```
>B10
@IF(@AND(A1=1,@OR(A2=2,A3=3)),A10,A11)
```

Note that the @OR arguments enclosed in parenthesis, is reduced to a 1 or a 0 before the @AND function operates. And finally. Here's one to keep you thinking for a while. If you have trouble following the logic, divide the outer @IF function into its three parts, reduce the first section to a 1 or a 0 by solving its sub arguments first. With the original numbers in column A, goto B10. Type:-

```
@IF(@OR(@IF(@AND(A1=1,A2=2),A4,A5)=(A3+1),A3>10,A4=8,A1=@PI),@I
F(@OR(@SQRT(A8+1)=A3,A1=1),A10,A11),@IF(A9<>10,@NA,A8^2))
```

Note that the right-hand side of an argument, =(A3+1), is enclosed in parenthesis, otherwise an error message will result. Change the variables and try to get @NA, @TRUE and @FALSE answers.

Relocating COMPDOS 1.2 by Rodney Eisfelder

In Response to Bob Best's query in last month's KAOS, I will try to explain how to get COMPDOS 1.2 working with any memory size.

COMPDOS 1.2 used to be supplied as several copies each designed for a different memory size.

For example, the 5 inch version had three copies for 24, 32 and 40K. Close study of these reveals that the only difference between them is in the high byte of absolute addresses of routines within COMPDOS. Therefore, to create a 48K version of COMPDOS, all that is needed is to identify these locations and apply the appropriate offset. This can be done permanently by using the extended monitor to make the 122 modifications required, saving the new version and making appropriate changes to BEXEC*.

Alternatively, you can get BEXEC* to make the changes at boot time, turning BEXEC* into a relocating loader! There are several stages required. The first listing is a program that compares the 40K and 32K versions of COMPDOS and generates DATA statements that are included in BEXEC* by issuing the statements:

```
DISK!"BS 06
DISK OPEN,6,"DATA"
DISK!"IO 20
```

Next you must modify BEXEC to do the relocation. The second listing is of selected lines from my version of BEXEC*. It is a bit messy, partly because it is necessary to shrink BASIC's workspace (line 1120) before making A$ (the address of COMPDOS).

Lines 1152 and 1153 load in the keyboard routine described in KAOS 3.8 and 3.9. Use of this also requires a few modifications to the DABUG part of COMPDOS.

The resulting COMPDOS/BEXEC* will work with any size memory as long as it is an even number of pages long. If the memory size is not 'standard', then booting takes a couple of seconds longer while BEXEC* relocates COMPDOS. The listing is for a 'standard' size of 40K, hence the 154 (=$9A) in line 1150.

If you have any problems, I can be contacted through the club or the Viatel Mailbox.

LISTING 1

```
75 REM setup start line and step
80 SB=110:ST=1
90 DISK!"SE B
100 DISK!"CA 4000=17,1
110 DISK!"CA 4600=16,1
115 DISK!"SE A
120 B1=16384:B2=B1+1536
130 DISK!"BS 06"
140 DISK!"CR DATA,01
145 DISK!"ZE DATA
```

```
150 DISKO,6,"DATA"
160 D=6
1000 L=0:N=SB:MX=60
1010 FORX=0TO1535:IFPEEK(B1+X)=PEEK(B2+X)THEN1100
1020 A$=MID$(STR$(X),2):IFL=0THENPRINT#D,N;"DATA";:N=N+ST
1030 L=L+LEN(A$)+1:PRINT#D,A$;
1040 IFL<MXTHENPRINT#D,",";:GOTO1100
1050 PRINT#D:L=0
1100 NEXTX
1105 IFL=0THENPRINT#D,N;"DATA";
1110 PRINT#D,"-1":PRINT#D,"EOF"
1120 DISKC,6
```

LISTING 2

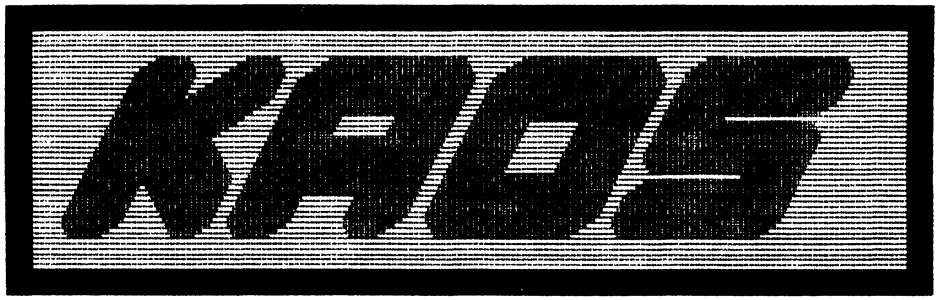
```
1 X=PEEK(10950):POKE8993,X:POKE8994,X
5 GOSUB600:GOSUB1100
100 READX:IFX)TTHENY=PEEK(BS+X):POKE(BS+X),Y+DF:GOTO100
110 DATA6,16,61,71,86,108,118,125,129,165,169,173,177,181,185,189,193
111 DATA197,201,207,210,218,241,246,299,318,346,349,384,410,413,431
112 DATA447,452,461,471,474,483,488,493,517,520,544,554,561,564,577
113 DATA603,636,644,652,667,736,840,857,869,922,930,938,941,949,1003
114 DATA1038,1048,1053,1056,1059,1062,1065,1072,1077,1080,1086,1091
115 DATA1100,1105,1108,1111,1114,1119,1128,1133,1136,1139,1146,1151
116 DATA1157,1160,1163,1166,1178,1189,1208,1214,1217,1220,1223,1232
117 DATA1243,1246,1249,1254,1259,1262,1268,1273,1277,1280,1283,1286
118 DATA1291,1330,1333,1354,1359,1362,1375,1387,1390,1395,1420,1431
119 DATA-1
145 BS=BS+1024
150 DISK!"GO "+A$:GOSUB1000:HM=HM-1
200 PRINT"LOADED ";(HM+7)/4;"K VERSION"


1100 PRINT" 1....OLD DOS":PRINT" 2....NEW DOS"
1110 INPUT"Selection ";S:IFS=1THEN510
1120 HM=PEEK(8960)-6:POKE133,HM:POKE8960,HM:CLEAR:HM=PEEK(8960)
1130 HM=HM+1:V1=INT(HM/16):V2=HM-V1*16+48:V1=V1+48:IFV1)57THENV1=V1+7
1140 IFV2)57THENV2=V2+7:A$=CHR$(V1)+CHR$(V2)+"00":BS=HM*256
1150 DISK!"CA "+A$+"=15,1":DF=HM-154
1152 DISK!"CA 3179=15,2"
1153 POKEBS+1365,49:POKEBS+1364,128
1155 IFDF=0THEN145
1160 T=0:GOTO100
```

Continued from page 5.

Prestel Terminal emulators suitable for Viatel are available for most micros including IBM PCs, Apples, Commodore 64s, and BBCs, but I am yet to see one for OSIs (note that the standard 24*24 screen on a Superboard is not good enough).

Those who insist on writing their own emulator and need a copy of the terminal specification should ring the Viatel Enquiries number listed in the 1985 phone book. Enquiries about subscribing to Viatel (which is NOT free) should be made to your local Telephone Business Office.

Eric Lindsay writes that he has recently obtained a copy of the Telecom Viatel specifications. He can supply a photocopy of the specifications for $3.50 (photocopying and postage). These specs should be available direct from Telecom, but his attempts to obtain them "officially" didn't work.
Address: E. Lindsay,



For People Who Have Got Smart

KIM-AIM-OSI-SYM-Apple-Mac-IBM-Rabble65-C64-Vic20-Peach-UK101

SURNAME_____CHRISTIAN NAME_____

ADDRESS_____

_____POST CODE_____

COUNTRY_____PHONE_____

FEES TO JANUARY 1986

$15 Australia, $20 N.Z. & P.N.G., $25 Europe & U.S.A.

My name and address may/may not be included in lists circulated to other KAOS members.

TYPE OF SYSTEM_____

AMATEUR RADIO?_____CALL SIGN_____RTTY_____

INTERESTS?_____

P.O.BOX 33, Bulleen, 3105. Ph.850 6420

KAOS-W.A.

Our last meeting had an attendance of 10 members and 4 computers which is quite good for us. We are still purely an O.S.I. group and look like stayng that way for the present. We also welcomed a new member Eric who has a Superboard hooked up to an electronic drum he is using in a band.

At this meeting discussion centered on the different Basic expansions people have such as Basic 5 or the single key commands offered in DABUG's.

At our next meeting Geoff Kendall hopes to bring his system and a modem. We will then be able to demonstrate how to access other computers and in particular the The Australin Beginning and see what programs for O.S.I. are available for us to load down and run. This is all dependant on Geoff having his modem up and running.

Our next meeting is on Sun 15th Sept at 2.00pm. It will be at our usual meeting place of GUILD HOUSE, 56 Kishorn Rd, Mt Pleasant.

See you there. GERRY LIGTERMOET

Registered by Australia Post
Publication No. VBG4212

If undelivered return to
KAOS, 24 Archibald St.
Pascoe Vale, VIC. 3044

KAOS KAOS
KAOS Postage paid Essendon Victoria Australia 3040 KAOS
KAOS KAOS